Optimization II Project I: Stochastic Programming

Peyton Lewis, Joe Morris, Vivek Dhulipalla, Rianna Patel
Group 16

Sp23 OPTIMIZATION II

Table of Contents

I. Introduction

In this project, our team was tasked with developing both a linear program and a quadratic program to provide alternative solutions to the standard newsvendor (NV) solution currently in use at the firm. Without concrete knowledge of the next day's demand, the newsvendor problem seeks to find the optimal quantity to print in order to maximize tomorrow's expected profit. Given a series of price and demand data for 99 days, we used the estimated parameters from the regression of demand on price as well as the corresponding residuals to develop our models. In both the LP and the QP, the publishing company is offered two avenues of flexibility in the form of quick-printing and disposal of the newspapers printed in excess of any given day's demand. In the standard NV model, these added benefits are unavailable, as this solution represents a simple approximation of reality. In an attempt to showcase the value of these flexibilities in generating profits above and beyond the simple newsvendor solution, we detail the comparison of model performance across the NV, LP, and QP results. Ultimately, the team discovered that implementing the QP program resulted in an increase in profit to the firm by approximately 5.30% at a relatively low computational cost. For these reasons, our team recommends its implementation for the firm's future operations.

II. Methodology

To begin this project, we utilized the demand and pricing data provided by the firm to fit a linear regression model to the data. Using demand as the dependent variable and price as the explanatory variable, we obtained the following regression result as shown in **Figure 1** below.

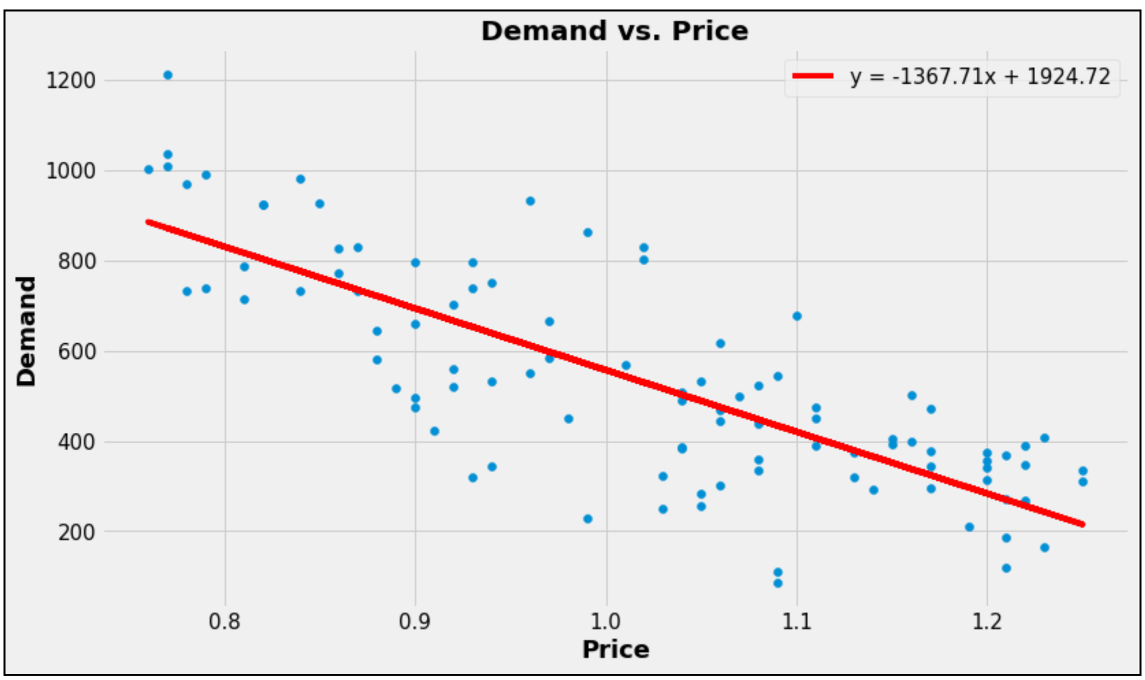

Figure 1: Scatter of Demand vs. Price and Regression Result

The coefficient estimate for the intercept (β_0) parameter was 1924.72 and the slope coefficient (β_1) was -1367.71. Using these coefficient estimates and the 99 residuals computed from the regression results, we were able to generate 99 new demand points for the given price of one dollar using the following formula as well as the subsequent code snippet:

$$New\ Demand_i = \beta_0 + \beta_1(p) + \varepsilon_i \quad for\ i = 1, 2, \ldots, 99$$

```
# generate new demand with p = 1
new_demand = intercept + (slope * p) + residuals
```

In the first extension of the NV problem when we develop the linear program, demand is not a direct function of price because price is fixed at $1.00. In the subsequent extension with our QP solution, demand will become a function of price. The details of this transformation will be covered in the QP section.

A. Linear Program

The benefit of the linear program lies in the flexibility it offers over the standard newsvendor solution. In an attempt to more closely reflect reality, the LP allows the printer to execute "quick print" orders in the event that he or she did not print enough newspapers to satisfy the day's demand. These rush orders cost g dollars per newspaper, and this cost, g, is in excess of the normal printing cost per newspaper, c. Additionally, if the firm were to print more than the demand on a given day, it would incur a disposal fee of t dollars per newspaper to get rid of the excess supply. It is important to note that in our model, all costs are positive. In light of these conditions, the firm would be able to exactly match any given day's demand with these capabilities. A summary of the parameters used in our models is provided in **Figure 2** below.

Parameter	Variable	Value
Selling Price	p	$1.00
Printing Cost	c	$0.50
Quick Print Cost	g	$0.75
Disposal Cost	t	$0.15

Figure 2: Parameters utilized in the Linear Program

Given these parameters, we sought to maximize the original objective function detailed below. In its initial formulation, the objective function found the optimal quantity of newspapers, q, that would result in the maximum expected daily profit for the next day, as measured by average daily revenue plus the negative average daily cost for the firm using the demand data. On days when the quantity of newspapers printed exceeded demand ($q > D_i$), the firm would face the disposal costs indicated by the term: $-t(q - D_i)$. Conversely, if the day's printed quantity was less than that day's demand ($q < D_i$), the firm would execute rush order printing to exactly meet that day's demand, as shown by the term: $-g(D_i - q)$.

In order to solve the problem, we needed to formulate the objective function as a linear program. To do so, we restructured the objective function in the manner shown below. We introduced dummy variables for each day, h_i, that represented part of the objective function. As can be seen, each value of h_i was set to be the negative cost on each day. Negative costs were utilized here because we were maximizing over h, so we wanted the negative costs to be as large as possible (i.e. closer to 0). As a result, the objective function maximized the average revenue plus the negative cost, where the revenue came from the pD_i term and the negative cost came from the h_i terms and the $-qc$ term.

Original Objective Function:

$$\max_{q} \frac{1}{n}\sum_{i=1}^{n}\left(pD_i - qc - g(D_i - q)^{+} - t(q - D_i)^{+}\right)$$

$$(x)^{+} = max(x, 0)$$

Rewritten Objective Function:

$$\max_{q,h} \frac{1}{n}\sum_{i=1}^{n}\left(pD_i - qc + h_i\right)$$

$$h_i = \; -g(D_i - q)^{+} - t(q - D_i)^{+}$$

$$(x)^{+} = max(x, 0)$$

Decision Variables:

- Optimal quantity: q
- Daily negative cost: $h_1, h_2, \ldots h_n$
- Total of $(n + 1)$ decision variables

Constraints:

$$h_i \leq -g(D_i - q)$$

$$h_i \leq -t(q - D_i)$$

$$h_i \geq -\infty$$

$$for\ i = 1, 2, \ldots, n$$

$$q \geq 0$$

As a vector with $(n + 1)$ entries for each of the decision variables: $[q, h_1, h_2, \ldots, h_n]$, the objective vector was created with the code below. The coefficient on the q decision variable was made to be $-c$ in order to account for the $-qc$ portion of the objective function. This is because in the objective function, $-qc$ is a constant that is not subscripted by i. The remaining coefficients on each of the h_i decision variables were each set to $\frac{1}{n}$ to allow the objective function to include the average negative cost.

```
# number of days
nd = demand.shape[0]

# objective
obj = np.zeros(nd+1)
obj[0] = -c # coefficient of q
obj[1:] = 1.0/nd # coefficient of the h's (to get the average negative cost)
```

The pD_i term in the beginning of the objective function is not included in the objective vector directly because it is a constant and is not multiplied by any of the decision variables. To include this value in the optimization algorithm and resulting objective value, we added the constant term $p * \overline{D_i}$ to the formula we used in the setObjective function in Gurobi when solving the model. In this term, D_i is the simulated demand we generated from the residuals (*new_demand*). The relevant code is shown below.

```
# solve the LP
lpMod = gp.Model()
lpMod_x = lpMod.addMVar(len(obj),lb=lb) # tell the model how many variables there are
# must define the variables before adding constraints
lpMod_con = lpMod.addConstr(A @ lpMod_x <= rhs)
lpMod.setObjective(obj @ lpMod_x + (p * np.mean(new_demand)),sense=gp.GRB.MAXIMIZE)
```

In terms of constraints, the model required two for each of the h_i decision variables, which resulted in a total of $2n$ constraints, where n was the number of days of demand data. As shown in the program summary above, there were $(n + 1)$ decision variables in the LP, which resulted in an A matrix with dimensions of $2n$ rows by $(n + 1)$ columns. For each day, the first constraint enforced the condition that $h_i \le -t(q - D_i)$ for all days. This resulted in a coefficient of t in the position of the quantity decision variable and a coefficient of 1 for the corresponding h_i decision variable in that row in the A matrix. Each of these constraints had a direction of $\le$ and a right hand side value of $t * new\_demand_i$, where the new_demand_i term was obtained by indexing the demand data generated from the residuals of the original regression when price was set to \$1.00. Likewise, the second constraint for each day was $h_i \le -g(D_i - q)$. This resulted in a coefficient of $-g$ in the position of the quantity decision variable and a coefficient of 1 in the corresponding position of that h_i decision variable in that row in the A matrix. Again, each of these constraints had a direction of $\le$ but a right hand side value of $-g * new\_demand_i$.

After setting the lower bound on the negative cost decision variables to be negative infinity and restricting the optimal quantity to a positive value, our model was finished. The code we utilized to loop through each day's demand data and create the A matrix is shown below.

```
# number of days
nd = demand.shape[0]

# rhs and direction
rhs = np.zeros(2*nd)
direction = np.array(['<']*(2*nd))

# constraint matrix
A = np.zeros((2*nd,nd+1))

# fill in the constraint matrix and rhs
for r in range(nd):
    A[2*r,[0,r+1]] = [t,1] # hi <= -t(q - Di)
    rhs[2*r] = t*new_demand[r]
    A[2*r+1,[0,r+1]] = [-g,1] # hi <= -g(Di - q)
    rhs[2*r+1] = -g*new_demand[r]
```

B. Quadratic Program

In our second extension of the newsvendor model, we allowed price to impact demand linearly with error according to the formula, $D_i = \beta_0 + \beta_1 p + \varepsilon_i$. As a result of this relationship, our quadratic programming model, unlike the linear model, was able to jointly solve for an optimal price in addition to an optimal quantity of newspapers to print. Like the LP, the model preserved the architecture for the quick print and disposal costs, but was no longer operating with the fixed price of $1.00. Similarly, this model relied on the initial regression output detailed at the start of the report with the firm's demand and price data. Using the initial β_0 and β_1 coefficient estimates, we were able to transform the objective function in the manner below.

As can be seen, the objective function bore a strong resemblance to the linear program's objective function in terms of its structure and use of the h_i decision variables for each day's negative cost. Likewise, the QP used the same decision variables [$q, h_1, h_2, \ldots, h_n$], but added another decision variable, p, to allow the model to optimize over price in its decision. Finally, in terms of constraints, the QP utilized the same $2n$ constraints for each day's negative cost (h_i), however, in place of each D_i term that was in the LP formulation, the following replacement term was used: $\beta_0 + \beta_1 p + \varepsilon_i$.

Objective Function:

$$\max_{p,q,h} \frac{1}{n}\sum_{i=1}^{n}(p(\beta_0 + \beta_1 p + \epsilon_i) - qc + h_i)$$
$$h_i = \; - g(D_i - q)^+ - t(q - D_i)^+$$
$$(x)^+ = max(x, 0)$$

Decision Variables:

- Optimal price: p
- Optimal quantity: q
- Daily negative cost: $h_1, h_2, \ldots h_n$
- Total of $(n + 2)$ decision variables

Constraints:

$$h_i \leq - g((\beta_0 + \beta_1 p + \epsilon_i) - q)$$
$$h_i \leq - t(q - (\beta_0 + \beta_1 p + \epsilon_i))$$
$$0 \geq h_i \geq - \infty$$
$$for\ i = 1, 2, \ldots, n$$
$$p, q \geq 0$$

In order to put the objective function into the code, we needed to formulate both the quadratic and the linear portions of the objective function in accordance with this general formula:

$$maximize\ x^T Q x + C^T x$$
$$s.t.\ constraints$$

In this formula, the x vector was the array of our decision variables [$p, q, h_1, h_2, \ldots, h_n$]. Using linear algebra, we determined the appropriate Q matrix and C vector for the problem using the series of computations detailed below.

To transform the objective function:

$$\max_{p,q,h} \frac{1}{n}\sum_{i=1}^{n} (p(\beta_0 + \beta_1 p + \epsilon_i) - qc + h_i)$$

$$\max_{p,q,h} \frac{1}{n}\sum_{i=1}^{n} p(\beta_0 + \beta_1 p + \epsilon_i) - \frac{1}{n}\sum_{i=1}^{n} qc + \frac{1}{n}\sum_{i=1}^{n} h_i$$

$$\max_{p,q,h} \frac{p}{n}\sum_{i=1}^{n} (\beta_0 + \beta_1 p + \epsilon_i) - \frac{1}{n}(n)\, qc + \overline{h_i}$$

$$\max_{p,q,h} \frac{p}{n}\sum_{i=1}^{n} (\beta_0 + \beta_1 p) + \frac{p}{n}\sum_{i=1}^{n} \epsilon_i - qc + \overline{h_i}$$

$$\max_{p,q,h} \frac{p}{n}(n)(\beta_0 + \beta_1 p) + p\overline{\epsilon_i} - qc + \overline{h_i}$$

$$\max_{p,q,h} p(\beta_0 + \beta_1 p) + p\overline{\epsilon_i} - qc + \overline{h_i}$$

$$\max_{p,q,h} p\beta_0 + p^2\beta_1 + p\overline{\epsilon_i} - qc + \overline{h_i}$$

Using this averaging logic, we successfully transformed the objective function into the general format used with quadratic programs. We separated it into its corresponding quadratic and linear components. The quadratic matrix accounted for the $p^2\beta_1$ term in our objective function, while the linear portion of the objective function handled the terms $p(\beta_0 + \overline{\epsilon_i})$, $-qc$, and $\overline{h_i}$. From here, we determined the appropriate coefficients for the decision variables in both the Q matrix and the C vector. They are illustrated in the matrix representation shown below:

$$[p \quad q \quad h_1 \quad h_2 \dots h_n] \cdot \begin{bmatrix} \beta_1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix} + \left[(\beta_0 + \overline{\varepsilon_i}) \quad (-c) \quad \frac{1}{n} \quad \frac{1}{n} \quad \dots \quad \frac{1}{n}\right] \cdot \begin{bmatrix} p \\ q \\ h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix}$$

When multiplied out, this matrix computation gave us the objective function that we desired, where the program maximized the average revenue plus the average negative cost of each day. Therefore, the objective value was a measure of the expectation of profit. The Q matrix was filled with zeros except for the upper left corner in which the β_1 coefficient was placed. The $x^T Qx$ term in the objective function generated the $p^2\beta_1$ term from the transformed objective function. The C vector was created by placing a coefficient of $(\beta_0 + \overline{\varepsilon_i})$ in the position for the p decision variable, a ($-c$) coefficient corresponding to the q decision variable, and coefficients of $(\frac{1}{n})$ in each of the remaining positions for the h_i daily cost variables. To implement these components in code, we took the following steps:

```
# number of decision variables
m = nd + 1 + 1

# Q matrix
Q = np.zeros((m, m))
Q[0, 0] = slope # upper left corner is beta 1

# linear component of objective
lin = np.zeros(m)
lin[0] = intercept + np.mean(residuals) # coefficient on p is beta 0 + mean of residuals
lin[1] = -c # coefficient on q is -c
lin[2:] = 1.0/nd # coefficient on the h's (to get the average negative cost)
```

Next, in order to handle the constraints, very few changes needed to be made from the constraints that were in place in the linear model. In a similar procedure, we looped through the 99 days of demand data, and instead of indexing that particular day's residual-generated demand value as we did in the LP, we simply replaced all instances of the D_i variable with the formula: $\beta_0 + \beta_1 p + \varepsilon_\text{i}$. This formula utilized the slope coefficient, the intercept, and the residuals from the initial linear regression model. Additionally, we utilized the same lower bound on the h_i decision variables of negative infinity, while restricting their upper bound to be zero. Finally, we restricted both the quantity (q) and price (p) decision variables to be bounded between zero and infinity. To implement these constraints in the QP model, we ran the following code:

```
# upper and lower bounds
ub = np.zeros(m)
ub[0:2] = np.inf # h's are zeros while p and q can be infinity
lb = np.zeros(m)
lb[2:] = -np.inf # lower bound on h's is -infinity

# rhs and direction of constraints
rhs = np.zeros(2*nd)
direction = np.array(['<']*(2*nd))

# constraint matrix
A = np.zeros((2*nd,m))

# fill in the constraint matrix and rhs
for r in range(nd):
    A[2*r,[0,1,r+2]] = [-t*slope, t, 1] # hi <= -t(q - (beta0 + beta1*p + e))
    rhs[2*r] = t*(intercept + residuals[r])
    A[2*r+1,[0,1,r+2]] = [g*slope, -g, 1] # hi <= -g((beta0 + beta1*p + e) - q)
    rhs[2*r+1] = -g*(intercept + residuals[r])
```

Using this structure, we solved the quadratic program and were able to obtain an optimal value for the price and quantity in order to maximize the expectation of the next day's profit for the firm. The results of the QP are detailed in the Results section of the report.

C. Bootstrap Samples - QP

After developing the model to solve the quadratic program, the team performed a series of bootstrap samples of the firm's demand and pricing data in order to evaluate the sensitivity of the optimal price and quantity to the dataset. The bootstrapping algorithm resampled the data with replacement, and created simulated demand data with the same number of records as the firm's original dataset. Sampling with replacement generates variations in the input data. In effect, any given bootstrapped sample can index an individual row several times, or not at all. The advantage of this technique was that it allowed our team to develop an estimate of the variability of our statistics (optimal *p, q, profit*) as well as construct confidence intervals around them.

In a series of 10,000 bootstrap samples, we solved the quadratic program 10,000 times and stored the values for each solution's optimal price, quantity, and profit. In each iteration of this procedure, we used the resampled data to run a new regression of the bootstrapped demand on price to obtain new coefficient estimates for the intercept and the slope term. To implement this procedure in our notebook, the following techniques were used:

```
# create a bootstrap sample of the data for 10000 samples
num_samples = 10000

# loop through the number of samples
for i in range(num_samples):

    # create a bootstrap sample of the data
    bootstrap_2 = np.random.choice(nd, nd, replace=True)
    boot_df_2 = demand.iloc[bootstrap_2]

    # get the regression output for the bootstrap sample
    slope_boot_2, intercept_boot_2, residuals_boot_2, model_boot_2 = get_regression_output(boot_df_2)

    # solve the QP for the bootstrap sample
    price_boot_2, quantity_boot_2, profit_boot_2 = solve_qp(slope_boot_2, intercept_boot_2, residuals_boot_2)
```

In one isolated iteration of the bootstrapping procedure, the following regression output was generated. As can be seen in **Figure 3** below, new measures of the intercept and slope coefficients were produced in the regression of the bootstrapped demand data on price. This procedure was repeated in each of the 10,000 bootstrap iterations. These coefficient results were then fed as direct inputs to the QP model. In each bootstrap of the dataset, the optimal price, quantity and expectation of profit were stored in order to analyze their variability.

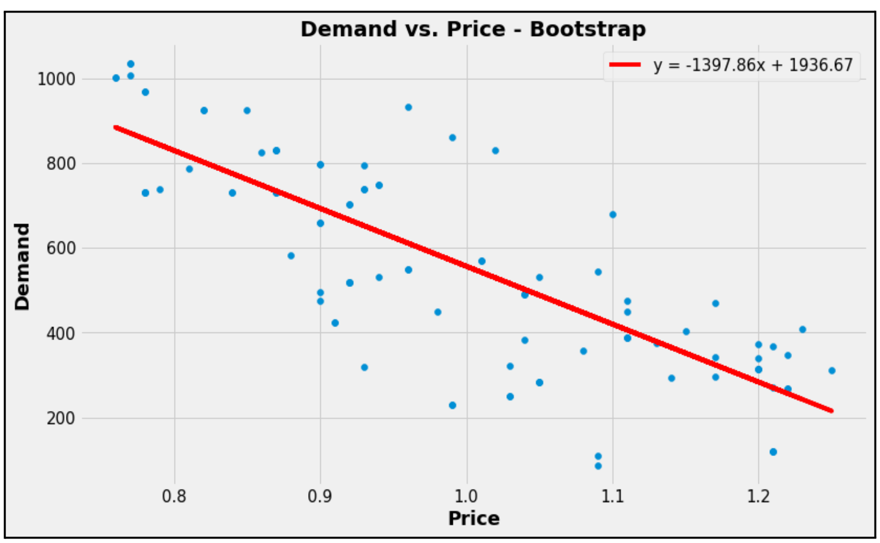

Figure 3: Sample Bootstrapped Regression of Demand on Price

To get a better sense of the variability around the mean of the bootstrapped results in the measures of the optimal price, quantity, and profit, we computed the mean, standard deviation, and the resulting 95% confidence interval around each of these variables. To generate a 95% confidence interval mathematically, the bootstrapping technique would be repeated an infinite number of times. In 95% of these iterations, the value of the unknown population parameter, in this case the population mean, would lie within the calculated confidence interval. To perform these confidence interval calculations, we used the following formulas and variable definitions:

$\bar{x}$ = sample mean
s = sample standard deviation
α = alpha value
df = degrees of freedom ($df = N - 1$)
N = number of bootstrapped samples

To determine the 95% Confidence Interval:

$$t_{critical} = t_{\alpha/2,\, df} = \pm 1.96$$
$$\text{for } \alpha = 0.05,\ N = 10{,}000$$

$$95\%\ \text{Confidence Interval} = \bar{x} \pm (t_{0.025,\, 9999}) * \frac{s}{\sqrt{N}}$$

As can be seen in the formula, a larger sample size N reduces the margin of error ($(t_{\alpha/2,\, df}) * \frac{s}{\sqrt{N}}$), the term added/subtracted from the sample mean to get the upper and lower bounds of each confidence interval. Because of this, a narrower confidence interval is constructed. With 10,000 bootstrapped samples, we obtained very narrow intervals, which gave more precise estimates of the unknown population mean for each of these variables. Therefore, having both a large sample size and data with low variability, we were able to produce estimates with less uncertainty.

III. Results

The high level overview of our various solutions to the newsvendor problem are shown in **Figure 4** below. We will take a more in-depth look at the results in this section.

	Price	Quantity	Profit
LP Solution	$1.00	471.87	$231.48
QP Solution	$0.95	535.29	$234.42
QP Bootstrap Solution	$0.95	535.78	$235.03

Figure 4: Comparison of Optimal Parameters (Price, Quantity, Profit) across LP, QP and QP Bootstrap Solutions

A. LP Solution

The linear program sought to maximize the expectation of the newsvendor's profit subject to the constraints defined in the Methodology portion of the report. As an improvement over the standard newsvendor program, the LP offered two primary flexibilities: quick printing and excess disposal. By restricting the price to be $1.00 and using the generated demand data, our team found that the optimal quantity of newspapers to print was approximately 472 newspapers as shown in **Figure 4**. This solution resulted in the maximum expectation of the next day's profit to be $231.48. For our analysis, the linear programming model is a good start to estimate the optimal profit, however, it is limited because this model fixes price and only allows an optimization over quantity.

B. QP Solution

As an improvement over the linear program, the quadratic program added the capability to optimize over price in addition to quantity. By expressing demand as a function of price, our QP determined the optimal price to sell the newspapers at was $0.95 and the optimal quantity to print was approximately 535 newspapers, as seen in **Figure 4**. Using these optimal parameters, the maximum expected profit was found to be $234.42, nearly $3.00 greater than the optimal profit obtained from the LP model. Given the added flexibility using p as a decision variable, the QP performed better than the LP, which was not a surprise. Allowing the firm to optimize over both price and quantity led to an even higher expected profit.

C. QP - Bootstrap Solution

After computing the optimal QP solution, we utilized bootstrap sampling methods to measure the sensitivity of the optimal price, quantity, and profit for this model. Looking at **Figure 4,** the mean values of each of the bootstrapped samples was taken to get the optimal results. To take a closer look at the bootstrapped results in the dimensions of the optimal price and the optimal quantity, we generated a scatter plot of all 10,000 samples with quantity plotted on the y-axis and price plotted on the x-axis. As shown in **Figure 5** on the following page, the scatter displayed a negative relationship between these two variables, as illustrated by the red trendline. Additionally, histograms of the optimal price and quantity data were added above and to the right of the corresponding axes. Looking first at the price histogram, it is clear that the majority of the optimal price values huddled between $0.94 to $0.96. In the optimal quantity histogram, the bulk of the bootstrapped values fell roughly between 520 and 540 newspapers.

Figure 5: Scatter Plot and Histogram of Bootstrapped Samples (Optimal Price and Quantity) - QP Solution

After exploring the scatter of the bootstrapped quantity and price values, we computed the following summary statistics to get a better sense of the variability around the mean for each parameter in the bootstrapped results. The respective means, standard deviations, and 95% confidence intervals for the optimal price, quantity, and profit are shown in **Figure 6** below.

	Mean	Standard Deviation	Lower 95%	Upper 95%
Price ($)	0.955	0.014	0.954	0.955
Quantity	535.78	32.87	535.14	536.43
Profit ($)	235.03	9.00	234.85	235.20

Figure 6: Summary Statistics for Optimal Price, Quantity, and Profit for the 10,000 Bootstrapped Samples - QP

Looking first at the results of the bootstrapping procedure on the optimal price, it can be seen that the mean value was approximately $0.95 with a standard deviation of $0.01. The 95% confidence interval around the average price was [0.954, 0.955], indicating that it had a very tight distribution. Moving next to the quantity variable, it can be seen that the mean value was 535.78 papers with a standard deviation of 32.87 papers. The 95% confidence interval around the mean was [535.14, 536.43], another tight confidence interval around the variable's mean. Finally, when looking at the distribution of the 10,000 bootstrapped profit values, it can be seen that the average profit was roughly $235.03 with a standard deviation of $9.00. The 95% confidence interval built around this mean was [234.85, 235.20]. Again, this represented an incredibly narrow margin around the average profit value. Taking a look at **Figure 7** below, the histogram of the 10,000 bootstrapped profit values is shown with the mean and standard deviation identified in the legend.

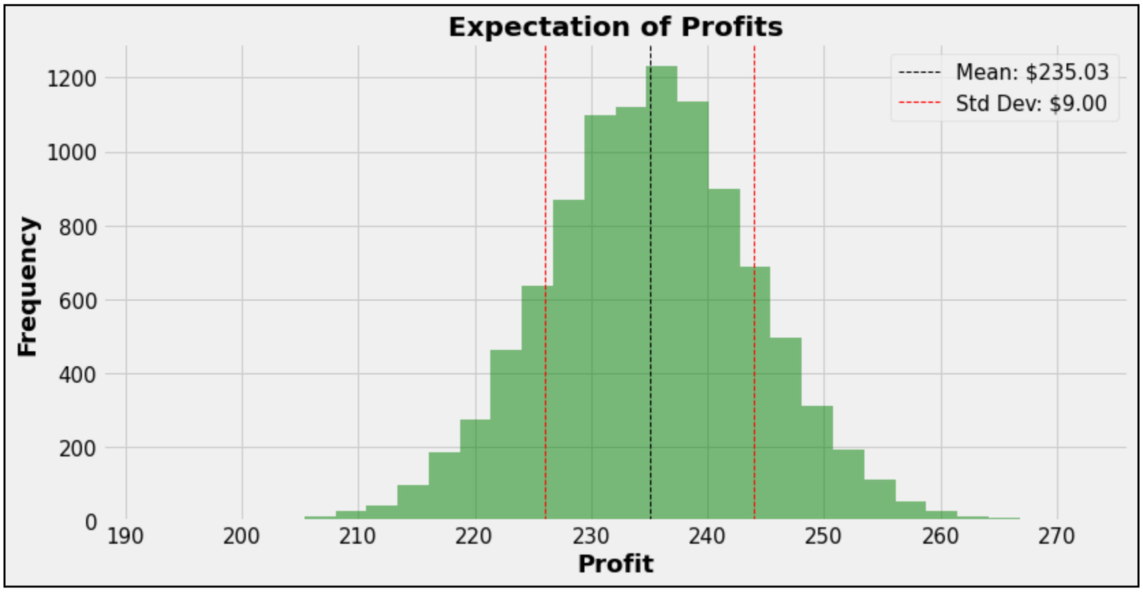

Figure 7: Histogram of Expectation of Profits over 10,000 Bootstrapped Samples - QP

By comparing the mean values of the bootstrapped price, quantity, and profit to the optimal QP solution (p = $0.95, q = 535.29, *profit* = $234.42), we saw consistent results. The original QP solution generated values of price and quantity that actually fell within their respective bootstrapped confidence intervals, while the QP optimal profit value fell just outside of its respective bootstrapped confidence interval.

The bootstrapped confidence intervals were extremely tight for the estimations of the parameters, therefore we had a fairly strong idea of the range in which the true population parameter for the price, quantity, and expected profit would lie. Since our QP solution was within the 95% confidence interval for price and quantity, and its optimal profit was just $0.43 away from being within its respective confidence interval, we were assured that our estimates of the parameters were close to their unknown population values. Ultimately, these findings supported the results of our original QP solution and led our team to conclude that the maximum expected profit generated by the QP model was representative of the true value of the expectation of profit. Therefore, the firm can be confident implementing our model in its operations.

D. NV Solution Comparisons

In order to draw comparisons between the various models that we constructed and the standard NV model, we made two distinct comparisons to it. Because the standard NV does not allow for any quick printing or excess disposal, we anticipated that it would produce suboptimal results to our more flexible models.

The first comparison was drawn between the linear program and the standard NV model. In the LP, like the standard NV, price was fixed at \$1.00 and the new demand data was used (generated from the residuals). In order to compare the LP output to the NV, we used the same price of \$1.00 and the same generated demand data in the newsvendor program. As can be seen in the results table in **Figure 8** below, the optimal quantity determined by the NV solution was 569.90 newspapers, which generated a profit of \$219.28. Because the LP generated an optimal expected profit of \$231.48, the solution of the NV model was inferior to the more flexible LP. This output shows that if the firm continues to run the NV program to determine the quantity of newspapers to print each day, it will miss out on approximately \$12.20 of profit by not implementing the LP model in its place. The reason the LP model offers a better solution than the NV model is a result of its flexibility in offering quick printing and excess disposals. The standard NV model is oversimplified and the firm is facing the opportunity cost of these forgone profits by not using a more advanced model.

When comparing the standard newsvendor model to the quadratic program, our team found a similar result. In order to compare these two models, we again fixed the price at \$1.00 and used the optimal quantity determined by the NV model in the comparison model explained above (q = 569.90). Using these parameters, we obtained the objective value associated with the QP model. Because the QP solution gave different values for the optimal price and quantity (p = \$0.95, q = 535.29), we knew that the corresponding objective value of the QP model with these suboptimal parameters would also be suboptimal. Using the fixed price and the same generated demand data from the regression output, we found the corresponding objective value in the QP model to generate \$222.63 of expected profit. Compared to the QP's optimal profit value of \$234.42, it is clear that the firm is leaving approximately \$11.79 of profit on the table. If deployed in place of the NV model, the quadratic model would allow the firm to capture a 5.30% increase in expected profit.

E. Model Performance

Using the results of the original LP and QP models as well as their NV model comparisons, we reached our conclusion regarding the optimal model for the firm to deploy. Taking a look at **Figure 8** below, it can be seen that the QP Bootstrapped Solution provides the firm the optimal expected profit of \$235.03 and the original QP Solution follows right behind it with an expected profit of \$234.42. In comparison to the linear programming model as well as both NV model comparisons, the QP model is optimal.

	Price	Quantity	Profit
LP Solution	\$1.00	471.87	\$231.48
NV Solution (LP Price \$1)	\$1.00	569.90	\$219.28
NV/LV Quantity and QP Objective	\$1.00	569.90	\$222.63
QP Solution	\$0.95	535.29	\$234.42
QP Bootstrap Solution	\$0.95	535.78	\$235.03

Figure 8: Comparison of Optimal Parameters (Price, Quantity, Profit) across all Models

IV. Conclusion and Recommendations

Although the standard newsvendor model provides the advantage of simplicity, it is built on the assumption that the selling price must be fixed and that there is no flexibility for engaging in quick printing or excess disposal to meet a given day's demand. When compared to the more advanced and realistic linear and quadratic programs, the standard NV solution was proven to be suboptimal, as it offered a lower expectation of profit in both of our comparison models.

The linear program resulted in an expected profit of $231.48 while the quadratic program produced an expected profit of $234.42. Clearly, both of these models improved upon the NV model profits of $219.28 and $222.63 from each comparison we drew. Out of all of the models that were tested, the QP model resulted in the highest objective value. We understand this to be the case because of the added flexibility this model offers above and beyond both the standard NV and the LP models. Because the QP allowed for the demand to be a function of price, the model possessed an increased level of versatility due to its ability to optimize not only quantity, but also price in order to generate the expected maximum profit for the firm. By giving the firm this added layer flexibility in addition to the quick printing and excess disposal capabilities, this model will allow the firm to more flexibly adapt to changing economic conditions in the future, should it need to. This is because the firm would not be forced to set a fixed price like it would be if either the standard NV model or the LP model were in use.

After evaluating the sensitivity of our QP model with 10,000 bootstrapped samples, we found that the narrow confidence intervals listed in **Figure 6** revealed that our QP solution estimates were minimally sensitive to variations in the firm's dataset. Additionally, implementing the QP model would come at a minimal computational cost to the firm, as the optimization algorithm solves just as quickly as the NV model. For these reasons, our team recommends that the publishing company implements the QP model in its future business operations to take advantage of its benefits and claim the extra profit extracted by this model.